

Docket No. AUS920010862US1

**APPARATUS AND METHOD OF CREATING A MIRRORING MAP FOR USE
WITH DIFFERENT PIECES OF DATA**

5

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to co-pending US Patent Application Serial No. _____ (IBM Docket No. AUS920011020US1), entitled APPARATUS AND METHOD OF REDUCING PHYSICAL STORAGE SYSTEMS NEEDED FOR A VOLUME GROUP TO REMAIN ACTIVE by the inventors herein, filed on even date herewith and assigned to the common assignee of this application.

BACKGROUND OF THE INVENTION

15

1. Technical Field:

The present invention is directed to a method and apparatus for managing data storage systems. More specifically, the present invention is directed to a method and apparatus for facilitating a volume manager to manage mirrored physical storage systems.

2. Description of Related Art:

Most computer systems are made up of at least one processor and one physical storage system. The processor processes, stores and retrieves data from the physical storage system under the guidance of an application program.

Application programs generally run atop an operating system. Among the many tasks of an operating system is that of allowing an application program to have a rather simplistic view of how data (i.e., data files) are stored within a physical storage system. Typically, an application

program views the physical storage system as containing a number of hierarchical partitions (i.e., directories) within which entire data files are stored. This simplistic view is often referred to as a logical view since most files are not 5 really stored as unit bodies into directories but rather are broken up into data blocks that may be strewn across the entire physical storage system.

The operating system is able to allow an application program to have this simplistic logical view with the help 10 of a file management system. The file management system stores directory structures, breaks up data files into their constituent data blocks, stores the data blocks throughout a physical storage system and maintains data logs of where every piece of data is stored. Thus, the file management 15 system is consulted whenever data files are being stored or retrieved from storage.

Computer systems that have a plurality of physical storage systems (e.g., servers) use an added layer of abstraction when storing and retrieving data. The added 20 layer of abstraction is a logical volume manager (LVM). Volume, in this case, is the storage capacity of a physical storage system. Thus, volume and physical storage system will henceforth be used interchangeably.

The LVM arranges the physical storage systems into 25 volume groups in order to give the impression that storage systems having each a much more voluminous storage capacity are being used. Within each volume group, one or more logical volumes may be defined. Data stored in a logical volume appears to be stored contiguously. However in 30 actuality, the data may be interspersed into many different locations across all the physical storage systems that make up the volume group.

Stated differently, each logical volume in a logical volume group is divided into logical partitions. Likewise, each physical volume in a volume group is divided into physical partitions. Each logical partition corresponds to 5 at least one physical partition. But, although the logical partitions in a logical volume are numbered consecutively or appear to be contiguous to each other, the physical partitions to which they each correspond, need not be contiguous to each other. And indeed, most often, the 10 physical partitions are not contiguous to each other. Thus, one of the many tasks of the LVM is to keep tab on the location of each physical partition that corresponds to a logical partition.

For fault tolerance and performance, some servers store 15 at least one extra copy of each piece of data onto the physical storage systems they use. Storing more than one copy of a piece of data is called mirroring the data. In order to store mirrored data, each logical partition used must correspond to as many physical partitions as there are 20 mirrors (or copies) of the data. In other words, if the data is mirrored three times, for example, each logical partition has to correspond to three physical partitions.

Presently, each time a system administrator decides that a piece of data is to be mirrored, the administrator 25 has to set up a mirroring map (i.e., which physical storage systems and what physical partitions in the physical storage systems to use to store the copies of the data). This can be a rather tedious task, especially, if the computer system is already using a mirroring map similar to what the 30 administrator intends to use.

What is needed, therefore, is an apparatus and method of allowing existing mirroring maps to be used whenever

Docket No. AUS920010862US1

applicable rather than defining a new mirroring map for each piece of data being mirrored.

SUMMARY OF THE INVENTION

The present invention provides a method, system and apparatus for allowing a mirroring map to be used for more than one piece of data. In other words, the invention allows a mirroring map that describes the layout of logical storage with respect a physical storage to be used with more than one piece of data. First, a mirroring map is created and given a name. Then, the name of the mirroring map and the mirroring map itself are stored into the computer system. When a piece of data is to be stored in accordance with the mirroring map any time thereafter, the name of the mirroring map is used to associate the mirroring map with the piece of data.

15

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The 5 invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

10 Fig. 1 is an exemplary block diagram illustrating a distributed data processing system according to the present invention.

Fig. 2 is an exemplary block diagram of a server apparatus according to the present invention.

15 Fig. 3 is an exemplary block diagram of a client apparatus according to the present invention.

Fig. 4 is a conceptual view of an LVM.

Fig. 5 illustrates a first example of a mirroring map that may be used with the present invention.

20 Fig. 6 illustrates a second example of a mirroring map that may be used with the present invention.

Fig. 7 illustrates an example of a piece of data mirrored in accordance with the mirroring map in Fig. 6.

25 Fig. 8 illustrates a flow diagram of a process used to create a mirroring map.

Fig. 9 illustrates a flow diagram of a process used to display mirroring maps.

Fig. 10 illustrates a flow diagram of a process used to associate a logical volume with a mirroring map.

30 Fig. 11 is a flow diagram of a process that may be used to ascertain that there is a requisite number of physical

storage systems available for a volume group to remain active.

Fig. 12 is a flow diagram of a process that may be used to determine whether a volume group has a requisite number 5 of physical storage systems available when a computer system is turned on or reset.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures, Fig. 1 depicts a pictorial representation of a network of data processing systems in which the present invention may be implemented. Network data processing system 100 is a network of computers in which the present invention may be implemented. Network data processing system 100 contains a network 102, which is the medium used to provide communications links between various devices and computers connected together within network data processing system 100. Network 102 may include connections, such as wire, wireless communication links, or fiber optic cables.

In the depicted example, server 104 is connected to network 102 along with storage unit 106. In addition, clients 108, 110, and 112 are connected to network 102. These clients 108, 110, and 112 may be, for example, personal computers or network computers. In the depicted example, server 104 provides data, such as boot files, operating system images, and applications to clients 108, 110 and 112. Clients 108, 110 and 112 are clients to server 104. Network data processing system 100 may include additional servers, clients, and other devices not shown. In the depicted example, network data processing system 100 is the Internet with network 102 representing a worldwide collection of networks and gateways that use the TCP/IP suite of protocols to communicate with one another. At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host

computers, consisting of thousands of commercial, government, educational and other computer systems that route data and messages. Of course, network data processing system 100 also may be implemented as a number of different 5 types of networks, such as for example, an intranet, a local area network (LAN), or a wide area network (WAN). Fig. 1 is intended as an example, and not as an architectural limitation for the present invention.

Referring to Fig. 2, a block diagram of a data 10 processing system that may be implemented as a server, such as server 104 in Fig. 1, is depicted in accordance with a preferred embodiment of the present invention. Data processing system 200 may be a symmetric multiprocessor (SMP) system including a plurality of processors 202 and 204 15 connected to system bus 206. Alternatively, a single processor system may be employed. Also connected to system bus 206 is memory controller/cache 208, which provides an interface to local memory 209. I/O bus bridge 210 is connected to system bus 206 and provides an interface to I/O 20 bus 212. Memory controller/cache 208 and I/O bus bridge 210 may be integrated as depicted.

Peripheral component interconnect (PCI) bus bridge 214 connected to I/O bus 212 provides an interface to PCI local bus 216. A number of modems may be connected to PCI local 25 bus 216. Typical PCI bus implementations will support four PCI expansion slots or add-in connectors. Communications links to network computers 108, 110 and 112 in Fig. 1 may be provided through modem 218 and network adapter 220 connected to PCI local bus 216 through add-in boards.

Additional PCI bus bridges 222 and 224 provide interfaces for additional PCI local buses 226 and 228, from which additional modems or network adapters may be supported. In this manner, data processing system 200 allows connections 5 to multiple network computers. A memory-mapped graphics adapter 230 and hard disk 232 may also be connected to I/O bus 212 as depicted, either directly or indirectly.

Those of ordinary skill in the art will appreciate that the hardware depicted in Fig. 2 may vary. For example, 10 other peripheral devices, such as optical disk drives and the like, also may be used in addition to or in place of the hardware depicted. The depicted example is not meant to imply architectural limitations with respect to the present invention.

15 The data processing system depicted in Fig. 2 may be, for example, an IBM e-Server pSeries system, a product of International Business Machines Corporation in Armonk, New York, running the Advanced Interactive Executive (AIX) operating system or LINUX operating system.

20 With reference now to Fig. 3, a block diagram illustrating a data processing system is depicted in which the present invention may be implemented. Data processing system 300 is an example of a client computer. Data processing system 300 employs a peripheral component 25 interconnect (PCI) local bus architecture. Although the depicted example employs a PCI bus, other bus architectures such as Accelerated Graphics Port (AGP) and Industry Standard Architecture (ISA) may be used. Processor 302 and main memory 304 are connected to PCI local bus 306 through PCI bridge 308. PCI bridge 308 also may include an integrated memory controller and cache memory for processor 302. Additional connections to PCI local bus 306 may be 30

made through direct component interconnection or through add-in boards. In the depicted example, local area network (LAN) adapter 310, SCSI host bus adapter 312, and expansion bus interface 314 are connected to PCI local bus 306 by direct component connection. In contrast, audio adapter 316, graphics adapter 318, and audio/video adapter 319 are connected to PCI local bus 306 by add-in boards inserted into expansion slots. Expansion bus interface 314 provides a connection for a keyboard and mouse adapter 320, modem 322, and additional memory 324. Small computer system interface (SCSI) host bus adapter 312 provides a connection for hard disk drive 326, tape drive 328, and CD-ROM drive 330. Typical PCI local bus implementations will support three or four PCI expansion slots or add-in connectors.

An operating system runs on processor 302 and is used to coordinate and provide control of various components within data processing system 300 in Fig. 3. The operating system may be a commercially available operating system, such as Windows 2000, which is available from Microsoft Corporation. An object oriented programming system such as Java may run in conjunction with the operating system and provide calls to the operating system from Java programs or applications executing on data processing system 300. "Java" is a trademark of Sun Microsystems, Inc. Instructions for the operating system, the object-oriented operating system, and applications or programs are located on storage devices, such as hard disk drive 326, and may be loaded into main memory 304 for execution by processor 302.

Those of ordinary skill in the art will appreciate that the hardware in Fig. 3 may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash ROM (or equivalent nonvolatile

memory) or optical disk drives and the like, may be used in addition to or in place of the hardware depicted in Fig. 3. Also, the processes of the present invention may be applied to a multiprocessor data processing system.

5 As another example, data processing system 300 may be a stand-alone system configured to be bootable without relying on some type of network communication interface, whether or not data processing system 300 comprises some type of network communication interface. As a further example, data
10 processing system 300 may be a Personal Digital Assistant (PDA) device, which is configured with ROM and/or flash ROM in order to provide non-volatile memory for storing operating system files and/or user-generated data.

15 The depicted example in Fig. 3 and above-described examples are not meant to imply architectural limitations. For example, data processing system 300 may also be a notebook computer or hand held computer in addition to taking the form of a PDA. Data processing system 300 also may be a kiosk or a Web appliance.

20 The present invention provides an apparatus and method that allow existing mirroring maps to be used for mirroring new pieces of data. Although the invention may preferably be local to server 104, it may nonetheless, be local to client systems 108, 110 and 112 of Fig. 1 or to both the
25 server 104 and clients 108, 110 and 112. Consequently, the present invention may reside on any data storage medium (i.e., floppy disk, compact disk, hard disk, ROM, RAM, etc.) used by a computer system.

30 To better understand the invention, a more detailed explanation of the LVM is needed. The LVM interacts with application programs and the physical storage devices as shown in Fig. 4. In Fig. 4 three layers are depicted, an

application layer 400, a logical layer 410 and a physical layer 420 each having one or more devices. It should be noted that the devices shown in the three layers are not all inclusive. There may be more devices in use in each of the 5 application layer 412, the logical layer 410 and the physical layer 430. Thus, the devices in Fig. 4 should be taken only as an example of devices that may be used.

The logical layer 410, for all intent and purpose, is the LVM. The LVM may be regarded as being made up of a set 10 of operating system commands, library subroutines or other tools that allow a user to establish and control logical volume storage. The LVM controls physical storage system resources by mapping data between a simple and flexible logical view of storage space and the actual physical 15 storage system. The LVM does this by using a layer of device driver code that runs above traditional device drivers. This logical view of the disk storage is provided to application programs and is independent of the underlying physical disk structure.

20 The logical layer 410 contains a logical volume 412 that interacts with logical volume device driver 414. A device driver, as is well known in the art, acts as a translator between a device and programs that use the device. That is, the device driver accepts generic commands 25 from programs and translates them into specialized commands for the device. In this case, the logical volume device driver 414 translates commands from an application program that may be executing on the computer system for device driver 430. Thus, when an application program sends 30 commands to file system manager 402 to store or retrieve data from logical volume 412, the file system manager 402 informs the logical volume manager 412 of the application

program's wish. The logical volume manager 412 then conveys the wish to the logical volume device driver 414. The logical volume device driver 414 then consults the appropriate map and instructs the device driver 430 which 5 ones of physical storage systems 422, 424, 426 and 428 to use for the data.

As mentioned in the Background of the Invention, when a system administrator wants to mirror a piece of data, the administrator has to devise a map (or mirroring scheme) to 10 correlate the logical volume being used to the actual physical storage systems in which the data is to be stored. Generally, this map includes the partitions of the physical storage systems that are to be used. As alluded to before, 15 this map is stored in the LVM. However, as the LVM is not aware that the physical storage systems are being used as mirrors of each other, the administrator has to maintain the map. For example, if one of the physical storage systems were to become defective and a replacement is used, the administrator has to update the map to include the 20 replacement. Furthermore, the administrator has to devise and maintain a map for each piece of data that is to be mirrored in the computer system whether or not there is already a map in the LVM that may be used for the new pieces 25 of data. Depending on the number of pieces of mirrored data in the computer system, this can be a rather tedious endeavor.

The invention uses an additional abstracted layer within a volume group in which physical storage systems may be defined as being mirrors of each other. Each mirrored 30 set is given a name. Thus, when a piece of data is to be mirrored in accordance with an existing map or mirroring scheme, the name of the set is associated with the logical

volume in which the data is to be written. If a replacement physical storage system is ever used, the LVM will automatically update the map.

Fig. 5 illustrates a first example of a mirroring map that may be used with the present invention. In this example, three physical storage systems (PSS) are used. The physical storage systems are PSS-1 500, PSS-2 510 and PSS-3 520. The three physical storage systems are divided into partitions (see partitions 502, 504 and 506 of PSS-1 500, partitions 512, 514 and 516 of PSS-2 510 and partitions 522, 524 and 526 of PSS-3 520). Mirrored data is stored in the three physical storage systems as illustrated by data A in partitions 502, 512 and 522 of PSS-1 500, PSS-2 510 and PSS-3 520.

Fig. 6 illustrates another mirroring map that may be used with the present invention. In Fig. 6, the physical storage systems are arranged in pairs. Each physical storage system of a pair contains a part of data A. For example, PSS-1 600 is paired off with PSS-4 610. Likewise, PSS-2 620 is paired off with PSS-5 630 and PSS-3 640 is paired off with PSS-6 650. PSS-1 600 contains the first half of data A and PSS-4 610 contains the second half of data A. The data in pairs PSS-2/PSS-5 and PSS-3/PSS-6 mirrors the data in pair PSS-1/PSS-4. This manner of storing data is called striping as a stripe of the data is stored in each physical storage system of a pair.

Obviously, both Fig. 5 and Fig. 6 illustrate examples of mirroring maps that may be used with the present invention. They, by no means, are all inclusive. In any case, the mirroring map of Fig. 5 may be named, for example, vgset₁ and that of Fig. 6 vgset₂.

Three new commands are used with the invention. The commands are: "create mirroring map", "display mirroring map" and "associate logical volume with mirroring map". In the commands, "mirroring map" and "logical volume" are 5 variables. For example, "mirroring map" may be the name of a particular map such as vgset₁. Logical volume is the name of a logical volume that an application program may use when storing or retrieving a data.

When the "create" command is used, it is mandatory that 10 the name of the mirroring map being created be provided. Likewise, when the "associate" command is used, both the name of the logical volume and the name of the mirroring map have to be provided. However, when the "display" command is used, it is not necessary that a name be provided. If a 15 name is not provided, all the mirroring maps that have thus far been defined and stored in the LVM will be displayed. This allows the administrator to choose which existing mirroring map to associate with a logical volume. If a name is provided with the associate command, only the mirroring map identified by the name will be displayed. Display of 20 the mirroring maps is as shown in Figs. 5 and 6.

As an example, suppose an application program is to store data "spam" into a logical volume called "logical volume₁" and suppose further that logical volume₁ is 25 associated with vgset₂ (see Fig. 6), spam will be stored in the physical volume set as shown in Fig. 7. That is, the first half of spam will be stored in partition 702 of PSS-1 700 and the second half of spam will be stored in partition 712 of PSS-4 710. Pair PSS-2 720 and PSS-5 730 and pair 30 PSS-3 740 and PSS-6 750 will contain a mirror image of pair PSS-1 700 and PSS-4 710.

In the example above, the format for the associate command may be: "associate logical volume₁ vgset₂ 1". Using this command, in essence, instructs the LVM to use mirroring map vgset₂ to store the data of logical volume₁.
5 Consequently, the data will be stored in physical storage systems PSS-1, PSS-2, PSS-3, PSS-4, PSS-5 and PSS-6 as shown in Fig. 7. The "1" after vgset₂ means that the first partition of the storage systems should be used. Specifying which partition to use is not necessary since without this
10 specification the LVM will use the next available partitions in the physical storage systems. Indeed, because of the high likelihood for errors associated with this option (i.e., forcing the LVM to use a partition that may already be used for another piece of data) only very sophisticated
15 administrators should ever use the option.

Fig. 8 is a flow chart of a process that may be used to create a mirroring map. The process starts when the create command is entered into the computer system (step 800). A check is made to determine whether a name is provided for
20 the mirroring map being created. If not, an error such as "name is needed" may be generated (steps 805 and 810). If a name is provided, then the user or administrator will be allowed to define the map. To do so, a display of a physical storage system may be provided.
25

Using copies (by cutting and pasting, for example) of the provided physical storage system, the administrator may make any mirroring map desired. After defining a map, the administrator has to specify which physical storage systems to be used by providing the name (each physical storage
30 system has a name) of the physical storage systems (e.g., PSS-1, PSS-2 etc.). Thus, the administrator may use two or more physical storage systems for one or more mirrored

copies of the data, and only one physical storage system for the other mirror or mirrors of the data or any combination thereof.

The administrator has to also specify how the data is
5 to be stored in the storage systems (whether the data should
be striped or not and what fraction of the data is to be
stored where, for example) by inserting the variable "A" or
"A/2" or "A/3" or "A/4" etc. into a partition of each
physical storage system. For example, suppose one copy of
10 the data is being stored across three storage systems and
A/2 is entered into one partition of a first storage system
and A/4 is entered into a partition of each of the other two
storage systems. Then, one half of the data will be stored
in the first storage system and one quarter of the data will
15 be stored in each of the other two storage systems. Note
that if "A" is entered, the data will not be striped or
rather the entire data will be stored into that storage
system.

If the administrator wants to stripe the data in three
20 stripes within only one physical storage system, the
administrator may enter "A/3" into three different
partitions of that storage system. Thus, the administrator
may instruct the LVM to mirror the data into the physical
storage systems in any combination the Administrator may
25 choose. When done, the name of the mirroring map and the
mirroring map itself are stored in the LVM for future use
(steps 805, 815 and 820).

Fig. 9 is a flow diagram of a process that may be used
to display the mirroring maps. The process starts as soon
30 as the display command is entered into the computer system
(step 900). A check is then made to determine whether a
name is provided. If so the mirroring map associated with

the name entered and the name of the mirroring map will all be displayed (steps 905 and 910). If not, all the mirroring maps stored in the LVM and their names will be displayed (steps 905 and 915).

5 Fig 10 is a flow diagram of a process that may be used to associate a logical volume with a mirroring map. The process starts when the associate command is entered into the computer system (step 1000). A first check is then made to determine whether a name of a logical volume is provided 10 with the command. If not, an error such as "a name of a logical volume is needed" may be generated (steps 10005 and 1010). If a name of a logical volume is provided with the command or after the name is entered, a second check is made to determine whether the name of the mirroring map is 15 provided with the command. If not, then an error such as "please enter a name of a mirroring map" may be generated (steps 10005, 1010, 1015 and 1020). If the name of a mirroring map is provided with the command or after entering the name of the mirroring map, a third check is made to 20 determine whether the mirroring map exists. If not, an error such as "mirroring map does not exist" may be generated (steps 1025 and 1030). If the mirroring map does exist, then the name of the logical volume provided is associated with the mirroring map and stored in the LVM for 25 future reference. That is, each time a piece of data is stored in the logical volume, the physical storage systems in the mirroring map will be used to store the data (steps 1025, 1030 and 1040).

One aspect of the invention allows a different quorum 30 to be used to ascertain whether or not metadata in the physical storage systems of a volume group is valid. As mentioned above, metadata is data about data. Metadata

describes how and when and by whom a particular set of data was collected, and how the data is formatted etc. Thus, with the use of metadata it can be determined whether or not a piece of data stored in a physical storage system is 5 stale. A piece of data is said to be stale if it has been updated and for some reason the update has not been stored in the physical storage system.

When an attempt to write metadata into a physical storage system is not successful, the physical storage 10 system is marked as unavailable. The present policy is that when half or more of the physical storage systems present that make up a volume group is marked unavailable, the volume group should not be used. This policy is used to ensure that when the computer system is turned on or is 15 reset if the majority of the physical storage systems in a volume group is available, the volume group may be used as metadata in at least one of the physical storage will be valid. The valid metadata may be copied into the other physical storage systems.

20 Using the present invention, if one of the physical storage systems out of each mirror set is available, then it can always be determined whether there is valid metadata. This reduces the number of physical storage systems for a volume group to remain active quite substantially.

25 As an example, suppose there are three (3) mirrors in each mirror set and a physical storage system is used for each mirror. Suppose further that there are thirty (30) mirror sets in a volume group. Then the number of physical storage systems in the volume group is ninety (90). 30 Presently, the number of physical storage systems needed for the volume group to remain active is 46 (i.e., a quorum consisting of the majority of the physical storage systems).

Using the present invention, however, only a quorum of thirty (30) physical storage systems is needed (one mirror out of each of the thirty (30) mirror sets).

Note that when the computer system is turned on or is 5 reset, a full mirror set and one mirror out of each of the remaining sets are needed to determine whether there is valid metadata. Using the example above, thirty-two (32) physical storage systems have to be available to determine whether there is valid metadata. This is so because of the 10 policy outlined above. That is, the minute that one of the mirrors that make up the thirty (30) mirrors becomes unavailable, the volume group should not remain active. This guarantees then that when the computer system is turned 15 on or is reset if there are at least a full mirror set and at least one mirror out of each of the remaining sets available it can always be determined whether there exists valid metadata in the volume group.

Fig. 11 is a flow diagram of a process that may be used 20 to ascertain that there is a requisite number of physical storage systems available for a volume group to remain active. The process starts when a computer system is turned on or reset (step 1100). A check is continuously being made 25 to determine whether all the physical storage systems in use are available. If not, a check will be made to determine whether at least one physical storage system from each of mirroring set is available. If so, the volume group will remain active. If not the volume group should not be used 30 anymore (steps 1105 - 1125)

Fig. 12 is a flow diagram of a process that may be used 30 to determine whether a volume group has a requisite number of physical storage systems available when a computer system is turned on or reset. Again the process starts when the

computer system is turned on or reset (step 1200). A check is made to determine whether all the physical storage systems in the volume group are available. If not, it is then determined whether an entire mirror set is available.

5 If not, the volume group may not activate (steps 1205 - 1220). If there is an entire mirror set available it will then be determined whether at least one physical storage system out of each of the remaining mirroring sets is available. If not the volume group may not activate. If so
10 the volume group may activate (steps 1225 - 1235).

The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations
15 will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with
20 various modifications as are suited to the particular use contemplated.